

REMARKS

Introduction

None of the claims have been amended herein. The application continues to include claims 1-21 and 25. Reconsideration of the rejection of the application is respectfully requested in view of the claim amendments and the following remarks.

The Outstanding Office Action did not Make a *Prima Facie* Case and is not Complete as to all Matters

With respect to the rejection of claims 1-21 and 25 under 35 U.S.C. § 103(a), the claimed features were essentially copied into the Office Action with citations and one or more terms appended to certain parts thereof, either without providing any reasoning or without matching the specific claimed features and portions of the cited art. For instance, in rejecting “receiving at execution time, a data processing specification having a first and a second data processing cell specification; specifying a first and a second data processing cell respectively, with each data processing cell specification having a plurality of statements including a formula specifying an action or computation”, the Office Action merely provided the statement “(e.g. *template match* - sec 7.1, pg. 28, 32; sec 7.6, pg. 35-36)”. Similar problems are present at least with respect to the rejections of the “analyzing” and “generating” features. For these features, the Office Action has not stated what discussion within the cited sections of “W3C”, which apparently comprises two different documents, is alleged to correspond with each claimed feature. In view of these and other shortcomings, Applicants

respectfully submit that the Office Action has not made a *prima facie* case of obviousness under 35 U.S.C. § 103(a), as is required by the MPEP and under U.S. law.

As stated in MPEP § 2142, “[t]he examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness.” “With regard to rejections under 35 U.S.C. 103, the examiner must provide evidence which as a whole shows that the legal determination sought to be proved (i.e., the reference teachings establish a *prima facie* case of obviousness) is more probable than not” (*Id.*). “The key to supporting any rejection under 35 U.S.C. 103 is the **clear articulation** of the reason(s) why the claimed invention would have been obvious. The Supreme Court in *KSR* noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made **explicit**” (MPEP § 2143, emphasis added). Further, MPEP § 2143.03 adds that, “**All words** in a claim must be considered in judging the patentability of that claim against the prior art” (emphasis added).

While citations to sections of the cited art were appended to the end of claimed features rejected in the Office Action, along with occasional terms and brief discussion, sufficient reasoning, and sometimes no reasoning, was not provided for many of the claimed features. Further, the Office Action does not indicate which specific features recited in the claims are believed to be taught by which sections of the cited art. For example, while “*template match*” was cited as allegedly corresponding with the claimed “receiving” features, the Office Action does not specify what, if any, specific parts of the cited art are alleged to teach, for example, “receiving at execution time”, “a first and second data processing cell specification”, etc. A rejection without proper reasoning is

inadequate since an Office Action lacking proper reasoning does not reasonably put the reader on notice as to **why** the Examiner rejected the claimed features. Rather, the reader is left to draw his or her own conclusions as to why the claims were rejected and accordingly, the burden has not been properly shifted to Applicants. Because the 35 U.S.C. § 103(a) rejection in the Office Action lacks a clear articulation of the reasons why the features recited in the claims would allegedly have been obvious in light of the cited art, the rejections cannot be supported per the requirements set forth by the United States Supreme Court and the Office Action is deficient.

Also, certain features do not appear to have been addressed at all. For example, the Office Action provided the following with respect to the rejection of claim 1: “generating one or more execution flow descriptions (e.g. *result tree* - sec 7, pg. 26-42; template which can be instantiated into a *result tree* - sec 1 Introduction, page 4)”. However, the features “describing the execution order of said plurality of statements of said first data processing cell specification based on results of the determination” were then included without any citation. Accordingly, the Office Action has not provided any indication as to where these features are allegedly found in the cited art. Accordingly, the Office Action is not “complete as to all matters” as required by 37 C.F.R. § 1.104(b).

Further, Applicants respectfully note that many of the page numbers in the citations do not match the page numbers of the referenced Internet documents available at the hyperlinks provided in the Office Action, making it difficult for Applicants to determine the location of broad sections that the Office Action cited. Additionally, while the XPath document was included in the rejection, because all sections cited with

respect to independent claim 1 are to section 7 or later, it does not appear that XPath was used in the rejection of this claim at all. Thus, the rejection as presented in the Office Action is incorrect.

If the Examiner continues to disagree with the reasoning presented below, Applicants respectfully request that the Examiner issue a new Office Action clarifying his positions, including specific citations and reasoning. In any case, the Office Action both fails to make a *prima facie* case under 35 U.S.C. § 103(a) and is not complete as to all matters. As such, the finality of the outstanding Office Action must be withdrawn.

The Claims are Allowable at least because the Cited Art Fails to Teach or Suggest Generating Execution Flow Descriptions based on Interaction or Computation References between Data Processing Cells, as Claimed

Claims 1-21 and 25 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over "XML Path language (XPath)" and "XSL Transformation (XSLT) Version 1.0 (W3C Recommendation 16 November 1999) in view of Renner et al. (U.S. Patent No. 6,993,657). Reconsideration of this rejection is respectfully requested at least because the cited art fails to teach or suggest generating execution flow descriptions based on interaction or computation references between data processing cells, as claimed.

One embodiment, as disclosed by Applicants, is a computer-implemented method of computing including receiving at execution time, a data processing specification having a first and a second data processing cell specification, unnested with respect to each other, specifying a first and a second data processing cell respectively, with each data processing cell specification having a plurality of

statements including a formula specifying an action or computation. The first data processing cell has a data dependency on the second data processing cell based on a reference to the second data processing cell in the formula that requires computation of the second data processing cell to evaluate the formula. The computer-implemented method also includes analyzing in real time the data processing specification, including the first and then the second data processing cell specification, to determine execution order of the plurality of statements specified by the first data processing cell specification, based at least in part on interaction or computation references between the first and second data processing cells. The computer-implemented method further includes generating one or more execution flow descriptions describing the execution order of the plurality of statements of the first data processing cell specification based on results of the determination. Additionally, the computer-implemented method includes, upon completion of the analyzing and generating, effectuating the data processing specified by the data processing specification in accordance with the execution flow descriptions.

In contrast to some embodiments of the present invention, none of the cited art teaches or suggests generating execution flow descriptions based on interaction or computation references between data processing cells, as claimed. Independent claim 1 specifically recites, in part, "analyzing in real time, the data processing specification including the first and then the second data processing cell specification, to determine execution order of said plurality of statements specified by said first data processing cell specification, based at least in part on interaction or computation references between

said first and second data processing cells" and "generating one or more execution flow descriptions describing the execution order of said plurality of statements of said first data processing cell specification based on results of the determination". In rejecting these features (with the exception of part of the "generating" as noted above), the Office Action cited the entirety of section 7, as well as sections 7.3, 7.6.1, 7.7 and 8 of the XSLT document, including the words "value=", "value of select" and "result tree" without providing any reasoning. Applicants respectfully submit that the cited art fails to teach or suggest these features.

XSLT generally discusses defining "the syntax and semantics of the XSLT language" which is "a stylesheet language for XML" (see Abstract, second paragraph, and section 1, first paragraph). "A transformation expressed in XSLT describes rules for transforming a source tree into a result tree. The transformation is achieved by associating patterns with templates" (section 1, second paragraph, of XSLT). "A stylesheet contains a set of template rules. A template rule has two parts: a pattern which is matched against nodes in the source tree and a template which can be instantiated to form part of the result tree" (section 1, fifth paragraph, of XSLT). "[T]he `xsl:value-of` element can be used to compute generated text, for example by extracting text from the source tree or by inserting the value of a variable. The `xsl:value-of` element does this with an expression that is specified as the value of the select attribute" (section 7.6, first paragraph, of XSLT). In other words, XSLT searches for patterns in nodes in a source tree and applies templates thereto. For example, the template:

```

<xsl:template match="person">
  <p>
    <xsl:value-of select="@given-name"/>
    <xsl:text> </xsl:text>
    <xsl:value-of select="@family-name"/>
  </p>
</xsl:template>

```

"creates an HTML paragraph from a person element with given-name and family-name attributes" (see section 7.6.1 of XSLT).

On the other hand, independent claim 1 recites generating execution flow descriptions based on interaction or computation references between data processing cells. As discussed with respect to some embodiments of the present invention with respect to Fig. 1, "a[n] x-sheet execution analyzer 122 is also advantageously provided to analyze the x-sheets 102, in particular, **determining the execution flow of their x-cells** 104, 'documenting' the flows in execution flow descriptions 132. Further, a[n] x-sheet execution engine 124 is provided to execute the x-cells 104 in accordance with the determined execution flow" (see, for example, page 5, lines 15-19, of the present specification, emphasis added). "Upon locating the next x-cell, analyzer 122 would determine if the located x-cell **references other x-cells**, either by way of the 'use' attribute, or by virtue of its formulas, block 204" (see, for example, page 14, lines 12-14, of the present specification, emphasis added). "[T]he cumulated interdependency information [is] ordered, block 210, and then output as **execution flow** 132, block 212" (see, for example, page 15, lines 1 and 2, emphasis added). Thus, as discussed in the specification and as claimed, interaction or computation references between data processing cells are determined and then these references are used to generate

execution flow descriptions. This differs from XSLT, which merely searches for patterns in nodes in a source tree and applies templates thereto, in essence performing a “find-and-replace” operation. XSLT is completely silent as to determining an execution flow where interdependencies exist among cells. Further, nothing is cited or found in XPath (not actually applied in rejecting claim 1) or Renner et al. that overcomes these deficiencies of XSLT.

Applicants also note that the Office Action's interpretation on page 10 is not permissible in light of the claims and specification. The Office Action alleged on page 10 that “[t]he language recited as ‘based on a reference to the second data processing cell in the formula that requires computation of the second data processing cell to evaluate the formula’ has been interpreted independent of the proffered ‘flow descriptors’ in the Specifications [sic], and is deemed matched by W3C cell specifications context (refer to more detailed explanation in the Rejection [no such detailed explanation actually present]) implicating the formula value-of select and the references or parameters found in parsing the formula.” However, as recited in claim 1, one or more **execution flow descriptions** describing the execution order of the plurality of statements of the first data processing cell specification are generated based on results of the determination. This “determination” clearly refers to determining the execution order of the plurality of statements specified by the first data processing cell specification based at least in part on interaction or computation references between the first and second data processing cells. Thus, generating an execution flow is based at least in part on interaction or computation references between data processing cells, as

explicitly recited in the claims and the interaction or computation references cannot be interpreted separately, as the Office Action alleges. Applicants note that execution flow descriptors are notably absent from XSLT.

In contrast to the cited art, independent claim 1 recites generating execution flow descriptions based on interaction or computation references between data processing cells. For at least the reasons discussed above, independent claim 1, and independent claims 11 and 21, which recite similar features but have a different scope, are allowable over the cited art. The remaining claims depend from independent claims 1 or 11 and add further features thereto. Accordingly, the dependent claims are also allowable over the cited art for at least the reasons discussed above with respect to the independent claims.

Conclusion

Applicants respectfully request favorable action in connection with this application.

The Examiner is invited and urged to contact the undersigned to discuss any matter concerning this application.

No fee should be required for this submission. However, should any fee be required, the Commissioner is authorized to charge any such fee to Counsel's Deposit Account 50-2222.

Respectfully submitted,

Date: August 31, 2009

/Michael A. Leonard II/
Michael A. Leonard II
Attorney for Applicants
Registration No. 60,180

Customer No. 74739

SQUIRE, SANDERS & DEMPSEY LLP
14TH Floor
8000 Towers Crescent Drive
Vienna, Virginia 22182-6212
Telephone: 703-720-7876
Fax: 703-720-7802

MAL:sjm